

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS E INFORMÁTICA
UNIDADE EDUCACIONAL PRAÇA DA LIBERDADE
Bacharelado em Engenharia de Software

Camila Silva Romagnoli
Leonardo Augusto Moreira Lourenço
Octávio Tabai Ribeiro Lage

RELATÓRIO: CARACTERÍSTICAS DE REPOSITÓRIOS POPULARES

Belo Horizonte
2022

Camila Silva Romagnoli
Leonardo Augusto Moreira Lourenço
Octávio Tabai Ribeiro Lage

RELATÓRIO: CARACTERÍSTICAS DE REPOSITÓRIOS POPULARES

Relatório apresentado na Sprint 1 da disciplina
Laboratório de Experimentação de Software.

Professor: José Laerte Pires Xavier

Belo Horizonte
2022

1 INTRODUÇÃO

O relatório apresentado tem como objetivo analisar as características de relatórios populares do GitHub. Essa atividade foi proposta na disciplina de Laboratório de Experimentação de Software.

1.1 Hipóteses

RQ01. Sistemas populares são maduros/antigos?

Métrica: idade do repositório (calculado a partir da data de sua criação).

Hipótese: O GitHub é uma plataforma de hospedagem de código fonte que foi criada em 2008. Portanto, um repositório pode ter no máximo 14 anos de idade. Será considerada a hipótese de que um repositório maduro tem aproximadamente a metade da idade do GitHub, ou seja um repositório maduro existe **há pelo menos 7 anos**.

RQ02. Sistemas populares recebem muita contribuição externa?

Métrica: total de pull requests aceitas.

Hipótese: Sistemas que recebem uma grande quantidade de contribuição externa são mais conhecidos. A hipótese levantada é de que repositórios populares têm a **quantidade de pull requests superior a 300**.

RQ03. Sistemas populares lançam releases com frequência?

Métrica: total de releases.

Hipótese: Sistemas populares geralmente possuem um grande volume de usuários e por este motivo lançam funcionalidades e melhorias com mais frequência. A hipótese será de que um repositório popular terá **pelo menos 3 releases por ano**.

RQ04. Sistemas populares são atualizados com frequência?

Métrica: tempo até a última atualização (calculado a partir da data de última atualização).

Hipótese: Sistemas populares geralmente recebem muitos feedbacks de usuários e passam por manutenções frequentes para correções e melhorias. Dessa forma, a hipótese é de que um repositório popular é atualizado **pelo menos uma vez por mês**.

RQ05. Sistemas populares são escritos nas linguagens mais populares?

Métrica: linguagem primária de cada um desses repositórios

Hipótese: As linguagens populares possuem comunidades grandes e ativas. Dessa forma, presume-se que repositórios nessas linguagens também sejam mais populares. De acordo com as linguagens mais populares apresentados no relatório [Octoverse](#), temos como hipótese que **Javascript, Python, Java, TypeScript e C# podem ser as linguagens mais utilizadas nos repositórios mais populares**.

RQ06. Sistemas populares possuem um alto percentual de issues fechadas?

Métrica: razão entre número de issues fechadas pelo total de issues

Hipótese: Sistemas populares são usados por uma grande quantidade de pessoas e por esse motivo estão sempre realizando manutenções para melhorar a usabilidade e corrigir falhas de forma rápida. Dessa forma, a hipótese é de que **pelo menos 75% das issues de um repositório popular devem estar resolvidas**.

RQ 07: Sistemas escritos em linguagens mais populares recebem mais contribuição externa, lançam mais releases e são atualizados com mais frequência?

Métrica: Divisão dos resultados obtidos nas RQs 02, 03 e 04 por linguagem e análise de como esses valores se comportam de acordo com as linguagem de cada repositório.

Hipótese: Linguagens de programação populares possuem alto engajamento da sua comunidade, ou seja, repositórios escritos em linguagens populares podem obter mais contribuições dessas pessoas. Dessa forma, foi levantada a hipótese de que **contribuições externas, quantidade de releases e frequência de updates são proporcionais a popularidade da linguagem**, visto que quanto mais pessoas são adeptas a ela, maior a comunidade e consequentemente mais pessoas para atuar nesses repositórios. Como referência para esta hipótese, assumimos que **para cada linguagem a quantidade de pull requests aceita deve ser pelo menos metade dos pull requests totais, a quantidade de releases por ano deve ser pelo menos 3 e a atualização deve ser pelo menos mensal.**

2 METODOLOGIA

Para realizar a análise das hipóteses, foi desenvolvido um script em Python com uma consulta em GraphQL que realiza consultas dos repositórios mais populares do Github e faz a exportação dos dados em um arquivo csv. O código-fonte do script e o arquivo csv está contido no repositório: https://github.com/octaviolage/ES_lab_VI_01/tree/main/sprint_01.

Foram analisados 1000 repositórios na data 20/02/2022 e calculadas as medianas através de fórmulas no Google Planilhas.

3 RESULTADOS OBTIDOS

RQ01: A mediana de idade dos repositórios é de 81 meses, que é aproximadamente 6,75 anos.

RQ02: A mediana de pull requests aceitas é de 411,5.

RQ03: A mediana de releases é 17. A mediana de quantidade de releases por ano de acordo com a mediana da idade dos repositórios é $17/6,75 = 2,5185$.

RQ04: A mediana do tempo entre atualizações é de 9 dias. Portanto, os repositórios são atualizados frequentemente.

RQ05: As 5 linguagens que mais apareceram nos repositórios populares foram: Javascript (240), Python (106), Typescript (97), Java (72) e Go (71). Vale ressaltar que em 113 dos 1000 repositórios analisados não foi identificada nenhuma linguagem de programação.

RQ06: A mediana da quantidade total de issues foi 1141 e a mediana de issues fechadas foi de 896,5. A proporção de issues fechadas para issues totais é igual a 78,57%.

RQ07: Resultados obtidos para cada uma das cinco principais linguagens dos repositórios analisados:

JavaScript

Mediana de **pull requests aceitas**: 450,5

Mediana **pull requests totais**: 724,5

Mediana de **releases** por ano: 3,0681

Mediana de **atualizações**: 11,5 dias

Python

Mediana de **pull requests aceitas**: 525

Mediana **pull requests totais**: 714

Mediana de **releases** por ano: 1,2173

Mediana de **atualizações**: 8 dias

Typescript

Mediana de **pull requests aceitas**: 1237

Mediana **pull requests totais**: 1808

Mediana de **releases** por ano: 14,0281

Mediana de **atualizações**: 3 dias

Java

Mediana de **pull requests aceitas**: 351

Mediana **pull requests totais**: 621,5

Mediana de **releases** por ano: 2,775

Mediana de **atualizações**: 7 dias

Go

Mediana de **pull requests aceitas**: 760

Mediana **pull requests totais**: 982

Mediana de **releases** por ano: 8,8941

Mediana de **atualizações**: 4 dias

4 ANÁLISE DOS RESULTADOS

RQ01: A hipótese inicial foi refutada, pois o resultado demonstrou que **a mediana de idade dos sistemas populares é de 6,75 anos**. No entanto, o valor encontrado foi próximo ao de 7 anos, que é o valor da hipótese. Portanto, sistemas populares não necessariamente são maduros.

RQ02: A hipótese inicial foi confirmada, pois o resultado demonstrou que **a mediana de pull requests aceitas foi de 411,5**. A hipótese era de 300 pull requests aceitas. Portanto, sistemas populares recebem muitas contribuições externas.

RQ03: A hipótese inicial foi refutada, pois o resultado demonstrou que **a mediana de releases ao ano é de 2,5185**. A hipótese inicial era de 3 releases ao ano. Portanto, sistemas populares não necessariamente lançam releases com frequência.

RQ04: A hipótese inicial foi confirmada, pois o resultado demonstrou que sistemas populares são atualizados **pelo menos a cada 9 dias**. Portanto, sistemas populares são atualizados frequentemente.

RQ05: A hipótese inicial foi parcialmente confirmada, pois o resultado demonstrou que **Javascript, Python, Java e Typescript figuram entre as linguagens primárias mais utilizadas nos repositórios populares**. No entanto, a linguagem C# que foi citada na hipótese não apareceu entre as 5 linguagens dos repositórios populares, enquanto a linguagem Go que não foi citada estava presente nos

repositórios mais populares. Foi identificado também que em segundo lugar do ranking (113) repositórios em que não foi identificada nenhuma linguagem de programação. Portanto, sistemas populares são escritos nas linguagens mais populares, mas não necessariamente repositórios populares são escritos em alguma linguagem de programação.

RQ06: A hipótese inicial foi confirmada, uma vez que **o percentual entre as medianas das issues fechadas e issues totais é de 78,57%, superando os 75% da hipótese inicial**. Portanto, sistemas populares possuem um alto percentual de issues fechadas.

RQ07: Javascript: A hipótese inicial para esta linguagem foi confirmada, pois repositórios em Javascript tem mais da metade (450) de **pull requests aceitas** em relação aos **pull requests totais** (724), possui mais **releases** por ano do que 3 (3,0681) e são **atualizados** pelo menos a cada 11,5 dias.

Python: A hipótese inicial para esta linguagem foi refutada, pois apesar dos repositórios em Python terem mais da metade (525) de **pull requests aceitas** em relação aos **pull requests totais** (714) e serem **atualizados** pelo menos a cada 8 dias, possuem menos **releases** por ano do que 3 (1,2173).

Java: A hipótese inicial para esta linguagem foi refutada, pois apesar dos repositórios em Java terem mais da metade (351) de **pull requests aceitas** em relação aos **pull requests totais** (621,5) e serem **atualizados** pelo menos a cada 7 dias, possuem menos **releases** por ano do que 3 (2,775).

TypeScript: A hipótese inicial para esta linguagem foi confirmada, pois repositórios em TypeScript tem mais da metade (1237) de **pull requests aceitas** em relação aos **pull requests totais** (1808), possui mais **releases** por ano do que 3 (14,0281) e são **atualizados** pelo menos a cada 2 dias.

C#: A hipótese inicial para esta linguagem foi refutada, pois ela **não aparece** entre as 5 linguagens mais utilizadas nos repositórios populares.

5 REFERÊNCIAS

GITHUB. **The State Of Octoverse**. Disponível em: <https://octoverse.github.com>.

Acesso em: 20 Fev. 2022.